

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Informaatika eriala

Janno Siim

Graafiteooria ülesannete komputeriseerimine

Bakalaureusetöö (6 EAP)

Juhendaja: dots. R. Prank

Tartu 2014

Graafiteooria ülesannete kompuuteriseerimine

Lühikokkuvõte: H. Pärn tegi 2013. aastal oma bakalaureusetöö raames prototüübi graafiteooria õpiprogrammist, millega oleks võimalik koostada ja lahendada erinevaid graafiteooriaga seotud ülesandeid. Antud lõputöö eesmärgiks sai seatud selle prototüübi edasiarendamine ja keerukamate ülesandetüüpide sissetoomine. Lisaks uuriti, milliseid ülesandetüüpe on üldse mõttekas sellises keskkonnas kasutada. Töö tulemusena realiseeriti 5 uut ülesandetüüpi ja lisaks täiendati keskkonda mitme olulise üldise täiendusega nagu näiteks suunatud graafide joonistamise võimalusega.

Võtmesõnad:

graaf, graafid, graafiteooria, õpiprogramm

Computerization of Graph Theory Exercises

Abstract: In 2013 in the context of his bachelor's thesis H. Pärn developed a prototype for educational software that would allow to create and solve different types of exercises related to graph theory. Current thesis sets its goal to develop that prototype and to introduce new and more complex exercise types. In addition, it is examined what types of exercises of graph theory can be computerized in the first place. As the result of this thesis 5 new exercise types were added to the software and also many general advances were made e.g. possibility to use directed graphs.

Keywords:

graph, graphs, graph theory, educational software, assessment software

Sisukord

Sissejuhatus	5
1 Probleemi kirjeldus	7
1.1 Aineist Diskreetse matemaatika elemendid	7
1.2 Ülevaade varasemast	7
1.2.1 Kasutajaliides	8
1.2.2 Eelnevalt realiseeritud ülesannete tüübid	10
2 Kompuuteriseeritavad ülesanded	12
2.1 Üldised mõisted	12
2.2 Euleri ja Hamiltoni graafid	14
2.3 Isomorfismid	15
2.4 Puud	16
2.5 Suunatud graafid	16
2.6 Relatsioonid	17
2.7 Mittekompoteeriseeritavad ülesandetüübid	18
3 Üldised täiendused	20
3.1 Juhuslik genereerimine	20
3.2 Mitme graafi paneel	22
3.3 Suunatud graafi paneel	23
3.4 Isomorfism	24
4 Realiseeritud ülesandetüübid	26
4.1 Regulaarsete graafide joonistamine	26
4.2 Sidusate komponentide joonistamine	27
4.3 Isomorfismi kontroll	30
4.4 Naabrusmaatriksile vastava suunatud graafi joonistamine	33
4.5 Suunatud graafide vastava naabrusmaatriksi konstrueerimine	34
Kokkuvõte	36

Summary	38
Kirjandus	38
Lisa 1 Ekraanitõmmised	41
Lisa 2 Litsents	46

Sissejuhatus

Tartu Ülikoolis on juba pikka aega tegeletud õpiprogrammide loomisega ning mitme informaatika teaduskonna aine puhul on need osutunud tõhusaks abivahendiks. Õpitarkvara abil on võimalik märgatavalt vähendada õppejõudude töökoormust, sest ajakuluka ülesannete kontrolli teostab arvuti. Samuti aitab ülesannete arvutis lahendamine muuta protsessi õpilaste jaoks mugavamaks, kuna õpitarkvara abil on võimalik sooritada mitmeid operatsioone, mille paberil teostamine tülikaks kujuneks. Lisaks on õpiprogrammi eeliseks õpilaste jaoks ka kohese tagasiside saamine.

Antud bakalaureusetöö peamiseks eesmärgiks on luua graafiteooria õpiprogramm Tartu Ülikooli informaatika teaduskonna aine "Diskreetse matemaatika elemendid" (edaspidi DME) jaoks. Heiki Pärn tegi 2013. aastal enda bakalaureusetöö raames graafiteooria ülesannete lahendamise keskkonna prototüübi [1], kuid see võimaldas vaid mõne suhteliselt elementaarse ülesandetüübi lahendamist. Siiski oli antud keskkonna loomine esimeseks oluliseks sammuks õpiprogrammi realiseerimisel ning on täiesti loomulik, et esmalt kompileeritakseki elementaarsed ülesandetüübid. Nüüd tuleks aga H. Pärna ellu viidud prototüüpi edasiarendada ja lisada juurde keerukamaid ülesandetüüpe, et keskkond oleks ka praktikas kasutatav.

Teiseks eesmärgiks on anda ülevaade, milliseid elementaarse graafiteooria ülesandeid on võimalik mõislikul viisil kompileerida. Rõhuasetus on just sellistel ülesandetüüpidel, mis on vastavuses aines DME õpetatava graafiteooriaga. Välja jäävad siinkohal kindlasti mingi väite tõestamist nõudvad ülesandetüübid, kuna töös käsitletav tarkvara on suunatud just erinevate graafide konstrueerimiseks. Ülevaade ülesandetüüpidest, mida saaks kompileerida, annab hea võimaluse kellelgi teisel tulevikus selle õpiprogrammi arendamist jätkata, kuna erinevaid ülesandetüüpe on ühe bakalaureusetöö raames realiseerimiseks liiga palju.

Kuna graafiteooriat õpetatakse ka mujal maailmas, võib tekkida küsimus ega analoogset õpiprogrammi juba loodud pole. Kuigi mõne seesuguse programmi

olemasolu pole välistatud, ei õnnestunud autoril siiski ühtegi sarnast tarkvara leida. Eksisteerib küll palju graafidega seotud programme, mis võimaldavad näiteks teha erinevaid teaduslikke arvutusi või näidata kuidas teatud algoritmide rakendamine graafil samm-sammult töötab. Kahe sellise programmi kohta võib lugeda H. Pärna tööst [1].

Teksti põhiosa on jaotatud neljaks peatükiks. Esimeses peatükis vaadeldakse lähemalt, miks selline tarkvara üldse vajalik on ja milline on H. Pärna loodud graafiülesannete keskkond. Teises peatükis on antud ülevaade ülesandetüüpidest, mida oleks võimalik programmi lisada. Kolmandas peatükis on kirjeldatud mõned üldised keskkonna täiendused nagu näiteks suunatud graafide joonistamise võimalus. Neljandas peatükis kirjeldatakse kõiki juurde lisatud ülesandetüüpe. Lisast 1 võib leida suure hulga ekraanitõmmiseid, mis aitavad programmi kasutajaliidesest paremini aru saada.

Peatükk 1

Probleemi kirjeldus

1.1 Ainet Diskreetse matemaatika elemendid

Graafiülesannete lahendamise keskkond on eelkõige mõeldud Tartu Ülikooli Informaatika teaduskonna aine Diskreetse matemaatika elemendid (edaspidi DME) jaoks. Siiski oleks antud tarkvara rakendatav ka mujal, kus tegeletakse elementaarse graafiteooria õpetamisega ning tõenäoliselt ei oleks ka programmi mõnda teise keelde tõlkimine väga mahukas töö.

Aine DME on suunatud eelkõige informaatikat ja arvutitehnikat esimest aastat õppivatele tudengitele. Aine eesmärgiks on anda informaatikas vajaminevad matemaatilised alusteadmised. Peamised teemad, mida selles aines tutvustatakse on lausearvutus, tõestamine, hulgateooria, relatsioonid ja graafiteooria.

Kuna tudengeid, kes ainet DME võtavad, on võrdlemisi palju, siis üritatakse aines juurutada õpiprogrammide kasutamist, mille tulemusel väheneb veidi õppejõudude töömaht, kuna lahenduste kontrollimine toimub automaatselt. Lisaks on õpiprogrammide eeliseks see, et teatud ülesandeid on programmi abil lahendada tunduvalt kergem kui paberil. Heaks näiteks DME kursuse raames käsitletavast valdkonnast, mille õpetamisesse on edukalt kaasatud õpiprogrammid, on lausearvutus.

1.2 Ülevaade varasemast

Järgnevalt antakse ülevaade sellest, milline oli graafiülesannete keskkond enne antud lõputöö käigus toimunud täiustamist. Ülevaade põhineb suures osas keskkonna autori Heiki Pärna bakalaureuse lõputööl [1] ja antud töö autori enda vaatlustel ja analüüsil.

Graafiülesannete keskkond koosneb kahest eraldiseisvast programmist: graafiülesannete koostamise programmist (edaspidi GKP) ja graafiülesannete lahendamise programmist (edaspidi GLP). Mõlemad programmid on kirjutatud keeles Java. GKP-l on kaks olulist eesmärki. Esiteks peab programmi abil olema võimalik koostada ülesannete kogusid ning teiseks peab sellega olema võimalik kontrollida lahendusfaile. See programm on suunatud õppejõududele. GLP peab võimaldama ülesannete kogudes olevate ülesannete lahendamist ja tulemuste salvestamist lahendusfaili, mida saaks GKP-ga kontrollida. See programm on suunatud eelkõige üliõpilastele.

1.2.1 Kasutajaliides

Graafiülesannete keskkonna kasutajaliidese puhul võib eristada programmispetsiifilisi paneele, andmestruktuuridega seotud paneele, ja andmestruktuuri paneele toetavaid abipaneele. Allpool on lühidalt kirjeldatud olulisimaid neist paneelidest. Detailsemat kirjeldust nende kohta võib lugeda H. Pärna bakalaureusetööst.

Programmisetsiifilised paneelid

Siia kuuluvad paneelid, mis on seotud kas ainult GKP-ga või ainult GLP-ga.

GKP ülesannete nimekirja paneel

See paneel näitab GKP-s antud ülesannete kogus olevat ülesannete nimekirja. Paneeli abil on võimalik avada, lisada ja eemaldada ülesandeid. Paneel avaneb nii uue ülesandekogu loomisel kui ka vana avamisel. (Lisa 1 - Joonis 10)

GLP ülesannete nimekirja paneel

Sellelt GLP paneelilt saab valida lahendamiseks ülesandeid. Lahendatud ülesandeid kuvatakse paneelil rohelise värviga, ebaõnnestunud ülesandeid punasega ning ülesandeid, mida ei ole veel proovitud lahendada, halliga. Paneel avaneb, kui avada GLP-s mõni ülesannete kogu. (Lisa 1 - Joonis 11)

Lahenduste kontrollimise paneel

See GKP paneel sisaldab kogu vajalikku infot lahendusfaili kontrollimiseks. Näha on lahendaja nimi ja ülesannete lahendamise katsed koos esinenud vigadega. Näidatakse ka seda, kas kontrollsumma on õige või mitte, st kontrollitakse, kas faili on üritatud käsitsi modifitseerida. Plussiga on märgitud lahenduskatsed, mis õnnestusid. (Lisa 1 - Joonis 12)

Andmestruktuuripaneelid

Andmestruktuuripaneelides hoitakse ülesande koostamiseks ja lahendamiseks vajalikke andmestruktuure nagu näiteks graafid ja maatriksid. Neid paneele kasutavad nii GKP kui ka GLP.

Graafipaneel

Graafipaneel on mõeldud suunamata graafide kuvamiseks ja konstrueerimiseks. Graafipaneeliga käib enamasti kaasas ka järgnevate nuppudega režiimipaneel: tipu lisamine, kaare lisamine, tipu kustutamine ja kaare kustutamine. Tipu lisamise režiimis on lisaks tippude lisamisele võimalik ka tippe liigutada ja neile nimesid anda. (Lisa 1 - Joonis 13)

Tähelepanu tuleks pöörata sellele, et edasiarenduse käigus on toimunud paar muudatust. Esiteks kasutatakse aines DME sõna “kaar” ainult suunatud graafide puhul ja sõna “serv” suunamata graafide puhul. Seega sai varasemalt mõlemale juhule viidanud sõna “kaar” suunamata graafide kohta käivates ülesannetes sõna “serv” vastu välja vahetatud. Teiseks tundus, et tipu liigutamine võiks olla välja toodud ka eraldi režiimima, et juhtida kasutaja tähelepanu selle funktsionaalsuse olemasolule. Samas jäi tipu liigutamise võimalus ka tipu lisamise režiimile alles, kuna juhul, kui kasutaja on sellest võimalusest juba teadlik, on mugavam tippe lisada ja liigutada ilma vahepeal mingile muule nupule vajutamata. Tipule nime andmine on võimalik mõlemas režiimis.

Maatriksipaneel

Maatriksipaneel võimaldab graafi esitust naabrusmaatriksi kujul. Selle paneeliga käib tihti kaasas ka nuppude paneel, mis laseb maatriksi järku suurendada ja vähendada. Juhul, kui maatriksi muutmine on lubatud, saab pärast rea või veeru aktiveerimist valida tipu nime ning maatriksi ruudule vajutades on võimalik määrata, kas tippudel vahel on serv (kaar). (Lisa 1 - Joonis 14)

Tipuastmejärjendipaneel

Viimane andmestruktuuride paneel võimaldab arvude järjendi konstrueerimist ja esitamist. Kaasas käib ka samasugune nuppude paneel nagu maatriksipaneelil, mille abil saab muuta järjendi pikkust. (Lisa 1 - Joonis 15)

Abipaneelid

Abipaneelid kujutavad endast andmestruktuuripaneele sisaldavad paneele, kuhu on täienduseks lisatud erinevaid nuppe ja lahtreid, mis on seotud konkreetse ülesandetüübiga. Abipaneele ei hakata siinkohal eraldi välja tooma, kuna peaaegu iga ülesandetüübi puhul on need veidi erinevad. Mitme ülesandetüübi puhul sisaldavad abipaneelid kahte andmestruktuuripaneeli. Kohati võivad veel konkreetse ülesandetüübi jaoks ka GKP ja GLP abipaneelid omakorda veidi erineda. Abipaneelide mitmekesisus on ka üks keerukuspunkte programmi täiendamise juures. Peaaegu iga uue ülesandetüübi lisamine eeldab vähemal või suuremal määral kasutajaliidese modifitseerimist, et nii ülesande konstrueerimine kui ka lahendamine oleks võimalikud.

1.2.2 Eelnevalt realiseeritud ülesannete tüübid

Järgnevalt antakse vaid põgus ülevaade sellest, mis oli kõnealuses õpiprogrammis antud bakalaureusetööle eelnevalt olemas. Tunduvalt põhjalikuma kirjelduse varasemalt realiseeritud ülesandetüüpidest võib leida H. Pärna tööst. [1]

Naabrusmaatriksile vastava graafi joonistamine

Ülesandes esitatakse naabrusmaatriks ja selle põhjal tuleb joonistada graaf. Kasutatakse graafi- ja maatriksipaneeli. Ülesande koostamisel GKP-s on võimalik muuta mõlema paneeli sisu. GLP-s on maatriksipaneeli osa lukustatud ning seega on seda võimalik näha, kuid mitte muuta.

Graafile vastava naabrusmaatriksi konstrueerimine

Tegemist on täpselt vastupidise ülesandega eelmisele. Esitatakse graaf ning selle põhjal tuleb koostada naabrusmaatriks. GKP-s on jällegi võimalik muuta nii graafi- kui ka maatriksipaneeli ja GLP-s on võimalik muuta ainult maatriksipaneeli.

Täisgraafi joonistamine

Siin ülesandes tuleb etteantud graafist moodustada täisgraaf. Kasutatakse ainult graafipaneeli. GKP-s määratud tippe ja servi ei saa ülesande lahendamise käigus eemaldada. Samuti ei saa lahendamise ajal lisada uusi tippe.

Täiendgraafi joonistamine

Selles ülesandes tuleb etteantud graafi põhjal moodustada täiendgraaf. Kasutatakse kahte graafipaneeli. Parempoolne graafipaneel sisaldab graafi, mille põhjal

tuleb moodustada täiendgraaf ning vasakpoolne graafipaneel sisaldab moodustavat täiendgraafi. GKP-s on mõlemad paneelid muudetavad. GLP-s on parempoolne paneel lukustatud.

Graafile vastava tipuastmejärjendi konstrueerimine

Ülesandes esitatakse graaf ning selle põhjal tuleb moodustada tipuastmejärjend. Seega kasutatakse graafi- ja tipuastmejärjendipaneeli. GKP-s saab muuta mõlemaid paneele ja GLP-s ainult tipuastmejärjendipaneeli.

Tipuastmejärjendile vastava graafi joonistamine

Tegemist on vastupidise ülesandega eelmisele ning sisult on see kuuest eelnevalt realiseeritud ülesandetüübist ilmselt kõige keerukam. Esitatakse tipuastmejärjend ning kui võimalik, siis tuleb selle põhjal joonistada graaf. Kasutatakse samu paneele, kuid GLP-sse on lisatud märkeruut, mida on võimalik valida juhul, kui lahendaja arvab, et ette antud tipuastmejärjendi põhjal ei ole graafi moodustamine võimalik. Kasutatakse taaskord graafi- ja tipuastmejärjendipaneeli. GLP-s ei saa tipuastmejärjendipaneeli muuta.

Kui eelmiste ülesannete puhul oli lahenduse kontrollimine suhteliselt triviaalne, siis selle ülesande kontroll nõuab veidi enamat. Et kindlaks teha, kas antud tipuastmejärjendi põhjal saab moodustada graafi, kasutatakse Erdős-Gallai teoreemi. [2, lk 5]

Peatükk 2

Komputeriseeritavad ülesanded

Järgnevalt on välja toodud suur hulk ülesannete tüüpe, mida saaks õpiprogrammi lisada. Teooria poolest vastavad ülesanded aine DME raames käsitletavale graafide teemale. Peamiseks ülesannete allikaks on aines DME kasutatav õpik “Diskreetse matemaatika elemendid” [3]. Lisaks sellele raamatule sai ülesandetüüpide jaoks ideid leitud ka A. Buldase, P. Laua ja J. Willemsoni raamatust “Graafid” [4], J. Matoušeki ja J. Nešetřili raamatust “Invitation to Discrete Mathematics” [5] ning O. Ore raamatust “Graafid ja nende kasutamine” [6].

Ülesannete valimisel arvestati kahte faktorit. Esiteks, kas ülesandetüüp on lihtsal viisil arvutis esitatav. Seejuures jäid koheselt kõrvale kõik ülesanded, mis nõudsid mingi väite tõestamist. Teiseks, kas ülesandetüüp on abiks õppematerjali omandamisel. Kohati on ülesannete juures antud ka põhjendusi, miks autor peab just sellist tüüpi ülesannet vajalikuks.

Suur hulk ülesandetüüpidest on esitatud parameetrilisel kujul, st ülesande kirjeldusse on sisse jäetud parameeter n, m, r vms. Seda tuleks mõista nii, et ülesannete kogu koostaja saab nendele parameetritele ülesandetüübi juures määrata konkreetseid arvulised väärtused. Kõik parameetrid on hulgast $\mathbb{N} = \{0, 1, 2, \dots\}$.

2.1 Üldised mõisted

Regulaarsed graafid

Ülesandes tuleb leida kõik n -tipulised regulaarsed graafid, kus n tähistab ette antud tippude arvu. Graaf tuleb esitada joonise või naabrusmaatriksina.

Intsidentsusmaatriks

Selles ülesandes tuleks graafi joonise või naabrusmaatriksi järgi koostada intsidentsusmaatriks. Samuti sobiks selle ülesande pöördülesanne, kus tuleks intsidentsusmaatriksi põhjal koostada graafi joonis või naabrusmaatriks.

Sidusad komponendid

Ülesandes tuleb esitada joonis graafist, millel on n tippu, m serva ja k sidusat komponenti, juhul kui see on võimalik. Lahendajale võib abiks tulla teoreem [3, lk 54], mille põhjal kehtib võrratus:

$$n - k \leq m \leq \frac{(n - k)(n - k + 1)}{2}$$

Joonisel 1 on üks võimalik lahendus ülesandele, kus küsitakse kuue tipu ja nelja servaga graafi, millel on kolm sidusat komponenti.



Joonis 1: Graaf, millel on 6 tippu, 4 serva ja 3 sidusat komponenti

Sillad ja eraldavad tipud

Ülesandeks on esitada joonis graafist, millel on täpselt m serva ja r silda. Teine antud ülesandega analoogne ülesanne kujutaks endast täpselt m serva ja r eraldava tipuga graafi joonise esitamist.

Täis- ja nullalamgraafid

Esitatakse graafi joonis ning ülesandeks on leida suurima tippude arvuga alamgraaf, mis on täisgraaf. Teine sarnane ülesanne oleks suurima tippude arvuga nullgraafi leidmine.

Lihtahel ja lihtsükkel

Selles ülesandetüübis tuleb graafi jooniselt leida pikim lihtahel ning teises sarnases ülesandetüübis tuleks leida pikim lihtsükkel.

Fikseeritud servade ja tippude arv

Ülesandeks on esitada kõigi omavahel mitteisomorfsete n tipuja m servaga graafide joonised.

2.2 Euleri ja Hamiltoni graafid

Euleri ja Hamiltoni tsükkel

Ülesandes esitatakse graafi joonis ning õpilane peab vastama küsimusele, kas tegemist on Euleri graafiga või mitte. Jaatava vastuse korral peab ta graafil ära märkima Euleri tsükli. Eitava vastuse puhul tuleb ära näidata tipp, mille tõttu ei ole tegemist Euleri graafiga. Ülesande eesmärgiks on see, et õpilane saaks tuttavaks Euleri graafi mõistega ning teaks tingimust, mille järgi Euleri graafi ära tunda. Sarnase ülesande saaks konstrueerida ka Hamiltoni graafide jaoks.

Euleri ja Hamiltoni graafide konstrueerimine

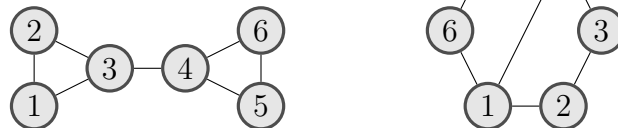
Tuleb konstrueerida n tipu ja m servaga Euleri graafi joonis. Ka Hamiltoni graafide jaoks saaks kasutada samasugust ülesannet.

Euleri ahel

Ülesandes esitatakse graafi joonis ning küsitakse, kas see sisaldab ahelat, mis läbib kõiki graafi tippe ja servi, kusjuures servi täpselt üks kord. Seda tüüpi ülesannete põhjal peaks õpilane jõudma järelduseni, et vastav ahel leidub parajasti siis, kui graaf on Euleri graaf või graafis on täpselt kaks paarituarvulise astmega tippu.

Hamiltoni graaf tipuastmete järgi

Ülesandes antakse ette graafi tipuastmete jada. Õpilasel tuleb tipuastmete järgi moodustada üks graafi joonis, mis sisaldab Hamiltoni tsükli, ja üks graafi joonis, mis ei sisalda Hamiltoni tsükli. See ülesanne näitab õpilasele, et erinevalt Euleri graafidest, pole ainult tipuastmete järgi alati võimalik kindlaks teha, kas tegemist on Hamiltoni graafiga või mitte. Joonisel 2 on üks näide sellise ülesande lahendusest.



Joonis 2: Vasakul mitte Hamiltoni graaf ning paremal Hamiltoni graaf tipuastmetega (3, 3, 2, 2, 2, 2)

2.3 Isomorfismid

Isomorfsuse kontroll

Ülesandes esitatakse kahe graafi joonised ning küsitakse, kas nad on omavahel isomorfsed. Programmi abil on võimalik graafi tippe liigutada, et paremini aru saada, kas graafid on isomorfsed või mitte. Isomorfsuse korral tuleb näidata, milline on isomorfne kujutus. Mitteisomorfsuse korral tuleb esitada põhjendus. Põhjendamist saaks realiseerida nii, et esitatakse teatud hulk invariante ning õpilane peab nende seast valima sellise, mis on mõlemal graafil erinev.

Näiteks sobiks selline komplekt invariante:

1. tippude arv;
2. servade arv;
3. komponentide arv;
4. minimaalse tsükli pikkus;
5. tipuastmejärjendid.

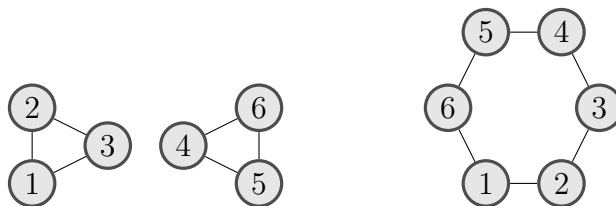
Tuleks arvestada ka, et võib leiduda graafe, mis on omavahel mitteisomorfsed, kuid ei erine invariantide komplekti poolest.

Isomorfsus täiendgraafiga

Ülesandes esitatakse graafi joonis ning küsitakse, kas see on isomorfne oma täiendgraafiga. Kõigepealt tuleb koostada antud graafi täiendgraafi joonis ning seejärel on võimalik vastata küsimusele.

Mitteisomorfsus tipuastmete järgi

Ülesandes esitatakse tipuastmete järjend ning selle järgi tuleb moodustada kahe graafi joonised nii, et need graafid ei ole isomorfsed. See ülesandetüüp näitab, et kaks graafi, millel on samad tipuastmed, ei pruugi ilmtingimata olla isomorfsed. Joonisel 3 on näide kahest sellisest graafist.



Joonis 3: Kaks mitteisomorfset graafi tipuastmetega $(2, 2, 2, 2, 2, 2)$

2.4 Puud

Puude konstrueerimine

Ülesandeks on konstrueerida kõigi n -tipuliste puude joonised.

Prüferi kood

Antakse ette Prüferi kood ning ülesandeks on selle järgi moodustada puu joonis. Samuti saab koostada vastupidise ülesande, kus puu joonise järgi tuleb moodustada Prüferi kood.

Puu kontroll

Ülesandes esitatakse graafi naabrusmaatriks ja selle järgi tuleb kindlaks teha, kas graaf on puu.

2.5 Suunatud graafid

Naabrusmaatriksi järgi suunatud graafi joonistamine

Ülesandes esitatakse suunatud graafi naabrusmaatriks ning selle põhjal tuleb koostada suunatud graafi joonis. Samuti saaks kasutada selle ülesande pöördülesannet, kus jooniselt tuleb minna üle naabrusmaatriksile.

Tugevalt sidusaks muutmine

Esitatakse suunamata graafi joonis. Küsitakse, kas graafile on võimalik määrata suunad niiviisi, et graaf oleks tugevalt sidus. Jaatava vastuse korral tuleb servadele määrata suunad nii, et graaf oleks tugevalt sidus. See ülesanne iseloomustab teoreemi, mis väidab, et graafi servadele saab määrata suunad nii, et graaf on tugevalt sidus parajasti siis, kui graafis puuduvad sillad.

Komponendid tugeva sidususe mõttes

Esitatakse suunatud graafi joonis ning ülesandeks on märkida ära kõik komponendid tugeva sidususe mõttes.

Suunatud Euleri graaf

Esitatakse suunatud graafi joonis ning küsitakse, kas see sisaldab suunatud Euleri tsüklit. Jaatava vastuse korral tuleb tsükkel ära märkida.

Suunatud Euleri ahel

Esitatakse suunatud graafi joonis ning küsitakse, kas see sisaldab suunatud ahelat, mis läbib kõiki graafi servi täpselt üks kord ja samas läbib ka kõiki graafi tippe.

2.6 Relatsioonid

Relatsioonidega seotud ülesanded on asjakohased, kuna üheks relatsiooni esituse viisiks on suunatud graaf. Samuti langevad kokku relatsioonimaatriks ja sellele relatsioonile vastava graafi naabusmaatriks. Seda küll juhul, kui mõlemas maatriksis vastavad samadele ridadele ja veergudele samad elemendid.

Omaduste kontroll

Esitatakse relatsioon mingil lõplikul hulgal X suunatud graafi joonise või maatriksi kujul. Küsitakse, kas sellel relatsioonil on teatud komplekt relatsiooni omadusi. Omadused, mille olemasolu kohta võidakse küsida, on refleksiivsus, antirefleksiivsus, sümmeetrilisus, antisümmeetrilisus, ja transitiivsus. Omaduste komplekti asemel võib ka küsida, kas relatsioon on ekvivalents, mitterange järjestus või range järjestus. Negatiivse vastuse korral oleks mõeldav ka kontranäite küsimine.

Relatsiooni konstrueerimine

Selles ülesandetüübis tuleks esitada n -tipulise suunatud graafi joonis, millel on teatud komplekt relatsiooni omadusi. Omadused on samad, mis eelmises ülesandetüübis. Ka seda ülesandetüüpi võiks rakendada maatriksi kujul relatsioonide puhul.

Sulundi leidmine

Esitatakse suunatud graafi joonis ning ülesandeks on täiendada graafi nii, et sellest saaks esialgse relatsiooni transitiivne sulund või refleksiivne transitiivne sulund. Sama ülesandetüüp oleks teostatav ka relatsiooni esitamisega maatriksi kujul.

2.7 Mittekomputeriseeritavad ülesandetüübid

Kontrastsuse mõttes tuuakse välja ka mõned ülesandetüüpide klassid, mis ei ole vähemalt selle programmi jaoks hästi kompoteeriseeritavad.

Tõestusülesanded

Suurem osa ülesannetest, mida graafiteooria õpikutest võib leida, on erinevate väidete tõestused. Arvestades, et tegemist on matemaatika haruga, siis on see küllaltki ootuspärane. Tõestusülesanded jäävad hetkel siiski selle programmi kontekstist täielikult välja. Praegusel kujul on programm suunatud just erinevatele konstrueerimisülesannetele. Samas ei taheta siin aga väita seda, et tõestusülesanded oleksid fundamentaalselt mittekompoteeriseeritavad. Nende ülesannete jaoks oleks vaja väga teistsugust lähenemist.

Tekstülesanded

Tekstülesanded on küll kindlasti mingil kujul kompoteeriseeritavad, kuid selleks, et oleks võimalik esitada mitmekülgseid graafiteooriaga seotud tekstülesandeid, tuleks autori hinnangul näha väga palju vaeva. Ilmselt on lihtsam jätta sellised ülesanded paberi peal lahendamiseks. Üks näide DME prakikumis kasutatavast tekstülesandest:

Infrastruktuuri parandamiseks otsustati riigi 16 suurema linna vahel rajada uusi maanteid. Valitsuse istungil võeti vastu kava, et lõpptulemusena peab kõigil linnadel peale pealinna olema otseühendus täpselt 2/3-ga kõigist ülejäänud linnadest, pealinnal aga peab olema otseühendus kõigi

ülejäänud linnadega. Kas selline kava on realiseeritav? Kui jah, siis tuua näide; kui, ei siis esitada põhjendus.

Vähese ülesannete arvuga ülesannete tüübid

Osad ülesandetüübid võivad sisalda vaid ühte või kahte erinevat ülesannet. Selliste ülesannete kompu­teriseerimine ei ole ilmselt väga tulus. Näiteks ei oleks mõtet kompu­teriseerida järgnevat DME praktikumis kasutatavat ülesannet:

Juurega puu sisaldab n tippu, kusjuures igal sisetipul (st tipul, mis pole leht) on parajasti k alamtippu. Leida puu sisetippude arv s ja lehtede arv l .

Antud ülesanne piirdub sellega, et sisestatakse kaks valemit.

Peatükk 3

Üldised täiendused

3.1 Juhuslik genereerimine

Kõigile ülesandetüüpidele sai lisatud juhusliku ülesande genereerimise võimalus, et suurendada erinevate ülesannete mitmekesisust ja vähendada aega, mis kuluks õppejõududel ülesannete manuaalseks sisestamiseks. GKP-s saab ülesannete kogu koostamisel ülesandetüüpide valiku aknas täita märkeruudu nimega “Konstrueeri ülesanne juhuslikult”, mille tulemusel ei tule ülesannete kogu koostajal käsitsi ülesannet komponeerida. GLP jaoks tähendab see seda, et iga kord, kui avatakse sellisel viisil implementeeritud ülesanne, genereeritakse vastavat tüüpi ülesande sisu juhuslikult.

Iga ülesandetüübi puhul tuleb määrata piirid, millisest hulgast võib ülesandeid genereerida. Näiteks ülesandetüüpide puhul, milles antakse ette graaf, tuleb määrata, kui palju võib olla tippu ning kui palju võib olla servi. Täpne piiride valik on olnud küllaltki subjektiivne, kuid autor on lähtunud sellest, et ka kõige minimaalsem ülesanne (nt vähima tippude ja servade arvuga graaf, mida on lubatud genereerida konkreetse ülesandetüübi jaoks) oleks mingilgi määral vaeva nõudev, samas kui maksimaalne ülesanne (nt suurima tippude ja servade arvuga graaf, mida on lubatud genereerida) ei oleks jälle liiga tülikas.

Nüüd kirjeldatakse, kuidas enamasti toimub juhusliku graafi genereerimise protsess. Antakse ette tippude arvu alampiir n_1 ja ülempiir n_2 , kus $n_1, n_2 \in \mathbb{N} \cup \{0\}$ ning $n_1 \leq n_2$. Hulgast $X = \{n_1 + 1, n_1 + 2, \dots, n_2 - 1, n_2\}$ valitakse graafi tippude arv n , kusjuures kõigil hulga X elementidel on võrdne esinemise tõenäosus ehk teisisõnu on tegemist diskreetse ühtlase jaotusega. Servade arvu vahemiku leidmiseks antakse ette kaks protsenti p_1 ja p_2 , kus $p_1 \leq p_2$. Servade arvu alampiiriks m_1 saab p_1 protsenti n -tipulise täisgraafi servade arvust ning ülempiiriks m_2 saab p_2 protsenti n -tipulise täisgraafi servade arvust. Sarnaselt valitakse nüüd diskreetse ühtlase jaotusega hulgast $\{m_1 + 1, m_1 + 2, \dots, m_2 - 1, m_2\}$

servade arv m . Juhusliku graafi saamiseks pannakse lõpuks kõik n -tipulise graafi servad massiivi, mis segatakse ära nii, et servad on juhuslikus järjekorras ning seejärel võetakse massiivi algusest m esimest serva. Sellega on juhusliku graafi genereerimise protsess lõppenud.

Juhusliku naabrusmaatriksi genereerimiseks kasutatakse sisuliselt samasugust protsessi. Erinevus seisneb vaid selles, et maatriksi puhul valitakse tippude arvu asemel maatriksi järk. Naabrusmaatriksi täitmiseks leitakse servad nagu juhusliku graafi genereerimisel, kuid seejärel esitatakse need servad naabrusmaatriksi kujul.

Edaspidi, kui ei ole just eraldi teisiti välja toodud, eeldatakse, et graafi ja naabrusmaatriksi juhuslikuks genereerimiseks on kasutatud eelkirjeldatud meetodeid. Seega märgitakse ära vaid tippude ja servade arvu alam- ja ülempiirid.

Järgnevalt antakse ülevaade, kuidas toimib juhuslik genereerimine kuue eelnevalt olemas olnud ülesandetüübi puhul. Uute ülesandetüüpide juhuslik genereerimine on kirjeldatud neile tüüpidele vastavates peatükkides.

Naabrusmaatriksile vastava graafi joonistamine

Genereeritakse järgmiste parameetritega juhuslik graaf:

- tippude arv on 4 kuni 7
- servade arv on 30% kuni 90%

Graafile vastava naabrusmaatriksi konstrueerimine

Genereeritakse järgmiste parameetritega juhuslik naabrusmaatriks:

- maatriksi järk on 5 kuni 7
- servade arv on 40% kuni 90%

Täisgraafi joonistamine

Genereeritakse järgmiste parameetritega juhuslik graaf:

- tippude arv on 4 kuni 7
- servade arv on 10% kuni 40%

Täiendgraafi joonistamine

Genereeritakse järgmiste parameetritega juhuslik graaf:

- tippude arv on 5 kuni 7
- servade arv on 40% kuni 80%

Graafile vastava tipuastmejärjendi konstrueerimine

Genereeritakse järgmiste parameetritega juhuslik graaf:

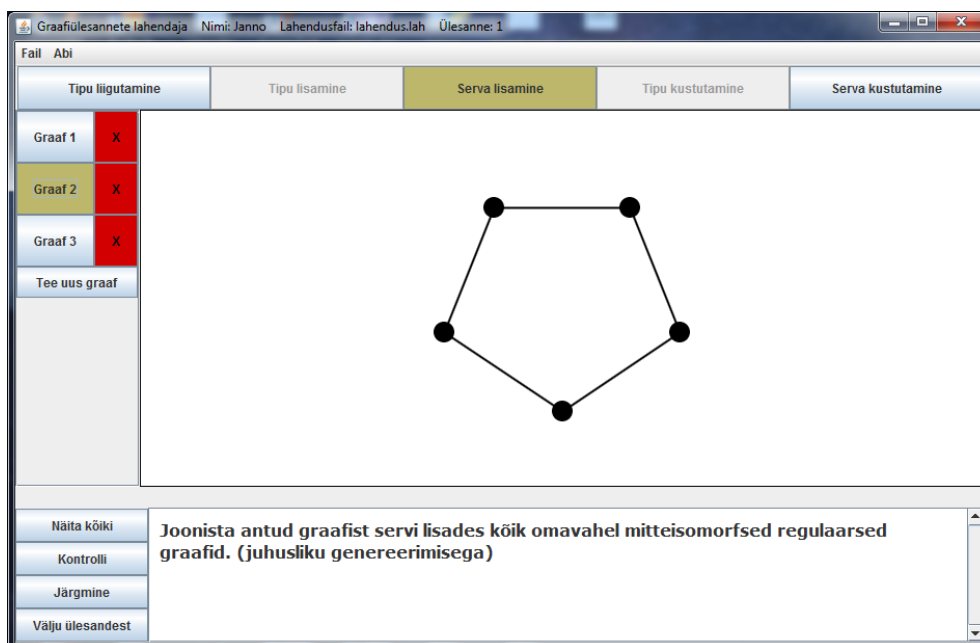
- tippude arv on 5 kuni 8
- servade arv on 40% kuni 90%

Tipuastmearjendile vastava graafi joonistamine

Genereeritakse juhuslik mittenegatiivsete täisarvudega järjend, mis on mittekasvav. Järjendi pikkus n valitakse ühtlase jaotusega vahemikust 5 kuni 8. Järjendi elemendid on valitud ühtlase jaotusega vahemikust 0 kuni $n - 1$. Lisaks on veel tehtud nii, et tõenäosusega $\frac{2}{3}$ on järjend selline, mille järgi on võimalik konstrueerida graaf ja vastavalt tõenäosusega $\frac{1}{3}$ ei ole võimalik järjendi järgi graafi moodustada. Selleks, et kindlaks teha, kas järjendi järgi on võimalik graafi joonistada, kasutatakse juba eelpool mainitud Erdős-Gallai teoreemi [2].

3.2 Mitme graafi paneel

Selleks, et oleks võimalik lisada ülesandetüüpe, kus tuleb joonistada suur kogus teatud omadustega graafe, oli vaja GLP-le abipaneeli, mis võimaldaks kasutajal suvalise arvu graafipaneelide haldamist samaaegselt (Joonis 4). Näiteks oleks see vajalik ülesandetüüpidele, kus tuleb joonistada kõik n -tipulised regulaarsed graafid või kõik n -tipulised puud. Edaspidi nimetatakse seda mitme graafi paneeliks.



Joonis 4: Kõigi 5-tipuliste regulaarsete graafide joonistamine. Hetkel on nähtaval regulaarne graaf astmega 2.

Antud paneeli vasakus servas on nupud, millega saab vahetada graafipaneele. Nuppude kõrval on punasel taustal olevad ristid, mille abil saab sellele nupule vastava graafi kustutada. Vajutades nupule “Tee uus graaf” avatakse uus graafipaneel ja pannakse see hetkel kasutusel olevaks graafiks. Uus graafipaneel sisaldab ülesannete kogust sisse loetud graafi, mis võib olla ka tühi.

Lisaks tuleb selle paneeliga veel kaasa võimalus näha kõiki joonistatud graafe korraga. Kui vajutada nupule “Näita kõiki”, siis avaneb aken, kus on kõik hetkel joonistatud graafid. (Lisa 1 - Joonis 16)

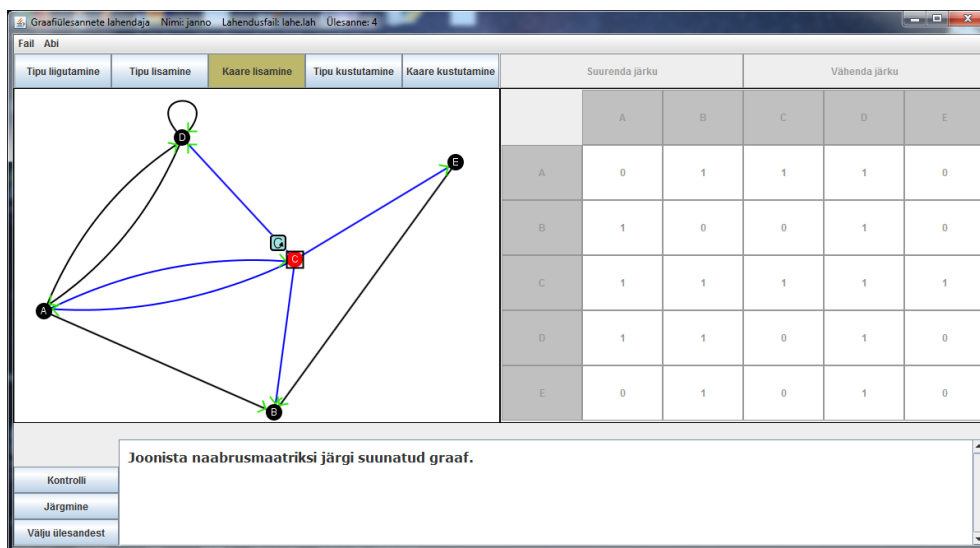
3.3 Suunatud graafi paneel

Kuna ülesandetüüpide otsimise tulemusel selgus, et eksisteerib arvestatav kogus suunatud graafidega seotud ülesandeid, mida oleks võimalik arvutil realiseerida, tekkis vajadus suunatud graafide esitamise võimaluse järgi. Selleks tuli luua suunatud graafi paneel, mis kujutab endast tavalise graafipaneeli edasiarendust.

Funktsionaalsuse poolest erineb suunatud graafipaneel tavalisest graafipaneelist vaid kaare lisamise režiimi poolest, kuid see-eest päris suurel määral.

Suunatud graafipaneelil on kolme eritüüpi kaari. Järgnevalt antakse nende kirjeldused, kusjuures kõiki neid tüüpe võib näha joonisel 5. Eeldatakse, et suunatud graafi paneel on kaare lisamise režiimis.

1. Olgu u ja v kaks sellist tippu suunatud graafipaneelil, et antud graafil ei eksisteeri kaari (u, v) ega (v, u) . Kui hiire vajutusega aktiveerida tipp u ja seejärel vajutada tipule v joonistatakse suunatud kaar tipust u tippu v . Kaar (u, v) on sirglõik, mille otsas on nool.
2. Olgu u ja v kaks sellist tippu, et leidub kaar (v, u) aga ei leidu kaart (u, v) . Kui nüüd samal viisil lisada kaar (v, u) , siis joonistatakse kaks kaart, mis kumerduvad teineteisest eemale.
3. Olgu w tipp millel puudub silmus. Kui aktiveerida tipp w , siis ilmub tipu ülemise vasaku nurga juurde väike ikoon, millele vajutades lisatakse kaarele w silmus. Silmuse suund on ülespoole, kui tipp w on suunatud graafi paneeli ülemises pooles ning silmuse suund on allapoole, kui tipp w on paneeli alumises pooles. Silmuse ikooni võib näha joonisel 5.



Joonis 5: Joonisel võib näha kõiki eriliiki kaari ja ka silmuse joonistamise ikooni.

3.4 Isomorfism

Leidub terve hulk graafiülesandeid, mis otsesel või veidi kaudsemal viisil vajavad isomorfismide kontrolli ning on ka igati loomulik, et graafiteooriaga seotud programm sisaldaks endas graafide isomorfismide kontrollimise võimalust. Graafide isomorfismide kontroll on ka kuulus probleem keerukusteoorias. Senini ei ole teada ühtegi üldist algoritmi, mis töötaks polünoomiaalse ajaga, see-eest leidub vihjeid, et sellist algoritmi ei eksisteeri [7, lk 1]. Polünoomiaalse ajaga algoritme leidub aga mitme olulise erijuhuga nagu näiteks puud [8], tasandilised graafid [9] ja tõkestatud tipuastmega graafid [10].

Esialgu oli küll plaanis isomorfismide kontrollimise jaoks kasutada mõnda teeki, kuid üllataval kombel ei õnnestunud Java jaoks leida ühtegi sobivat. Seega tuli autoril endal realiseerida isomorfismide kontrollimise algoritm. Kasutusele sai võetud algoritm VF2, mida oli võrdlemisi lihtne realiseerida ning mis töötab kiiremini, kui konkureerivad algoritmid. [11] Kuigi VF2 ei tööta polünoomiaalse ajaga, osutus see siiski piisavalt kiireks selles õpiprogrammis kasutatavate väikese tipuarvuga graafide jaoks.

Antud õpiprogrammis realiseeritud versioon algoritmist VF2 ei järgi päris täpselt L. P. Cordella, P. Foggia, C. Sansone ja M. Vento artiklis "An Improved Algorithm for Matching Large Graphs" [11] antud VF2 kirjeldust. Esiteks on algoritm realiseeritud suunamata graafide jaoks, kuigi seda ei oleks väga keeruline ka suunatud graafide jaoks mugandada. Teiseks ei tee realisatsioon hetkel kindlaks, kas üks graaf on isomorfne teise graafi alamgraafiga, kuigi täielik VF2 realisatsioon

seda tegelikult võimaldaks. Kuna sellist funktsionaalsust ei olnud aga realiseeritavate ülesannete jaoks vaja, jäeti aja kokkuhoiu mõttes see võimalus hetkel välja.

Peatükk 4

Realiseeritud ülesandetüübid

Nüüd vaadeldakse lähemalt ülesandetüüpe, mis antud lõputöö raames programmis realiseeriti. Iga ülesandetüübi juures on kirjeldatud GKP ja GLP kasutajaliides ja funktsionaalsuses, ülesande lahendamise käigus tekkivad võimalikud süntaktilised ja sisulised vead ning ülesandetüübi juhusliku genereerimise protsess. Süntaktilisteks vigadeks loetakse sellised vigu, mis ei ole otseselt seotud ülesande sisuga ja nende tegemise puhul ei loeta ülesannet valesti lahendatuks. Täpselt vastupidi on aga sisuliste vigadega. Kõigi vigade puhul ilmub vastavasisuline veateate aken.

4.1 Regulaarsete graafide joonistamine

Ülesandes sisuks on ette antud tippude arvu n järgi joonistada kõik omavahel mitteisomorfsed regulaarsed n -tipulised graafid.

GKP-s saab selle ülesande valimisel määrata joonistatavate graafide tippude arvu. Tippude arv peab kuuluma vahemikku 0 kuni 15. Ülesannete sooritamise seisukohast on mõistlikud tippude arvud vahemikust 4 kuni 7. Suurema tippude arvu korral on erinevaid regulaarseid graafe õpilase teadmiste kontrollimiseks juuba põhjendamatu palju. Näiteks 8 tipu korral tuleks joonistada 22 erinevat graafi.

GLP kasutab selle ülesande puhul mitme graafi paneeli. Iga uus graaf täidetakse automaatselt vajaliku arvu tippudega. Iga kord, kui kasutaja soovib luua uut graafi, kontrollitakse, kas kõik seni loodud graafid on regulaarsed ja omavahel mitteisomorfsed. Kui selgub, et kaks graafi on omavahel isomorfsed, teavitab programm, millised need joonistatud graafidest olid. Samuti kui leidub graaf, mis ei ole regulaarne, annab programm teada, millise graafiga on tegu. Alles siis, kui mõlemad tingimused on korrektselt täidetud, lubatakse teha uus graaf.

Kui kasutajale tundub, et kõik n -tipulised regulaarsed graafid on joonistatud, saab ta vajutada kontrollimise nupule ning programm teavitab, kas ülesanne on

tõepoolest õigesti lahendatud. Kontrollimiseks kasutatakse väga lihtsat meetodit. Nimelt on programmi sisse kirjutatud, kui palju on regulaarseid graafe iga lubatud tipuastme korral. [12]

Süntaktilised vead puuduvad.

Sisulised vead:

1. joonistatud graaf ei ole regulaarne;
2. joonistatud on kaks graafi, mis on omavahel isomorfsed;
3. mõni regulaarne graaf on joonistmata.

Juhusliku genereerimise puhul valitakse tippude arv ühtlase jaotusega vahemikust 4 kuni 7.

4.2 Sidusate komponentide joonistamine

Ülesandeks on moodustada n tipu, m serva ja k komponendiga graaf, juhul kui see on nende parameetrite korral võimalik.

Kuna antud ülesanne ei nõua koostamise osas graafide ega muu sarnase konstrueerimist, tundus olevat mõistlik teha muudatusi ülesannete valiku aknas niiviisi, et selle ülesandetüübi valimisel oleks koheselt võimalik sisestada ka vajalikud parameetrid. Kui GKP-s valida ülesannete nimekirjast antud ülesanne, siis muutub aken suuremaks ning joone alla tekib ülesande kirjeldus ning tippude, servade, ja komponentide arvu sisestamise lahter nagu on näidatud joonisel 6.

Joonis 6: Komponentidega graafi konstrueerimise ülesandetüübi parameetrite sisestamise aken

Sisendi puhul kehtivad järgmised kitsendused:

1. tippude arv n ei tohi olla suurem kui 15;
2. komponentide arv k ei tohi olla suurem kui n .

Servade arvu puhul ei ole otseselt takistavat kitsendust, kuid kui servade arv m ei rahulda võrratust

$$n - k \leq m \leq \frac{(n - k)(n - k + 1)}{2} \quad (*)$$

siis antakse hoiatus, et antud graaf ei ole joonistatav.

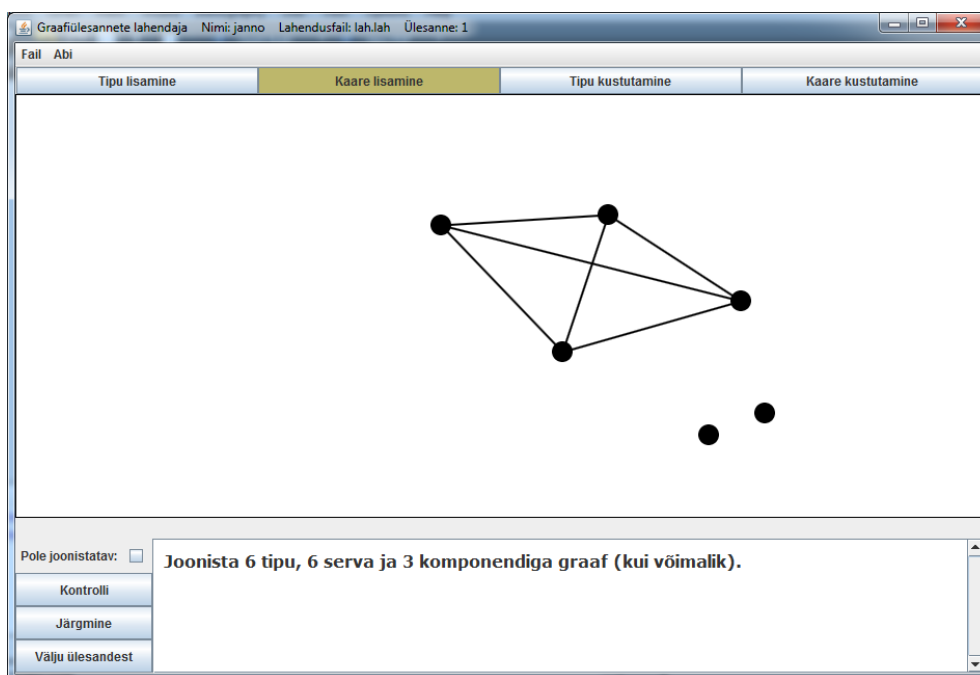
Lahenduse programmis kasutatakse selle ülesande jaoks graafipaneeli, kuhu on juurde lisatud märkeruut "pole joonistatav". (Joonis 7)

Kontrolli puhul tehakse kõigepealt kindlaks, kas ülesanne on lahendatav. Selleks kasutatakse võrratust (*). Sidusate komponentide kindlaks tegemiseks kasutatakse graafi süvitsi läbimist, et teada saada, millised tipud on omavahel ahelaga ühendatud.

Süntaktilised vead puuduvad.

Sisulised vead:

1. vale tippude arv;



Joonis 7: Komponentide ülesande GLP vaade.

2. vale servade arv;
3. vale komponentide arv;
4. märgitud on, et graaf ei ole joonistatav kuigi tegelikult on joonistatav.

Juhusliku genereerimise korral valitakse tippude arv n ühtlase jaotusega vahemikust 6 kuni 9. Sidusate komponentide arvu alampiir on 2 ja ülempiir valitakse vastavalt tippude arvule järgmiselt:

Tippude arv n	6	7	8	9
Komponentide arvu ülempiir	2	3	3	4

Olgu valitud sidusate komponentide arv k . Tõenäosusega $\frac{3}{5}$ on ülesanne lahendatav, tõenäosusega $\frac{1}{5}$ on servi liiga vähe ja tõenäosusega $\frac{1}{5}$ on servi liiga palju. Kui ülesanne peab olema lahenduv, siis valitakse servade arv võrratusega (*) fikseeritud vahemikust. Kui servi peab olema liiga vähe, siis valitakse servade arv m vahemikust $2 \leq m < n - k$. Kui servi peab olema liiga palju, siis valitakse servade arv m vahemikust $\frac{(n-k)(n-k+1)}{2} < m \leq \frac{n(n-1)}{2}$.

4.3 Isomorfismi kontroll

Selles ülesandes antakse ette kaks graafi ning õpilase ülesandeks on näidata, milline on nende kahe graafi vaheline isomorfism või põhjendada, miks need graafid ei ole isomorfsed.

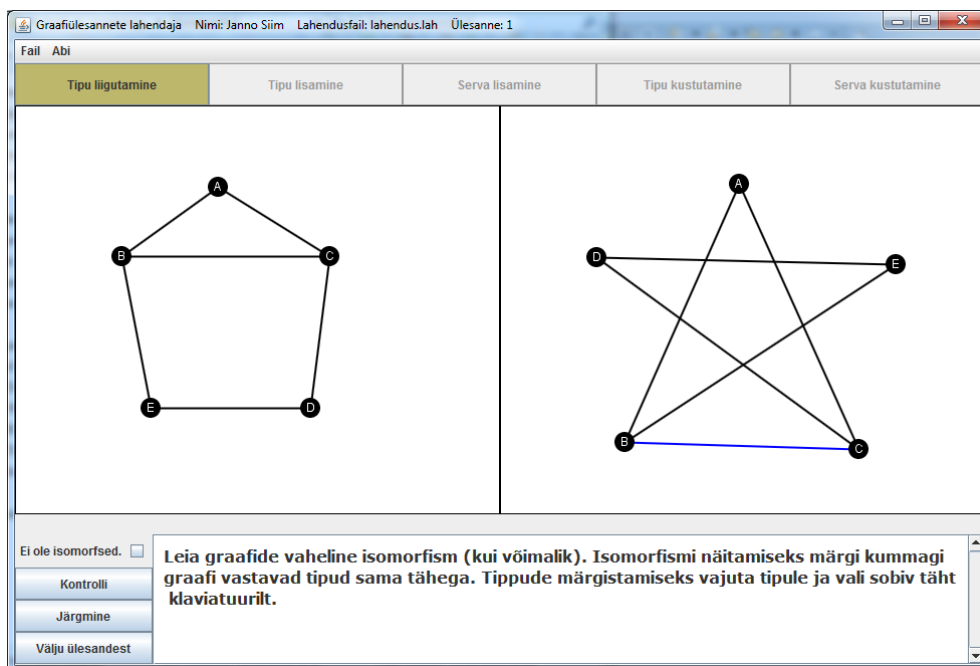
Ülesannete koostamise programmi pool on selle ülesandetüübi jaoks üpriski ootuspärane. Valides ülesandetüüpide nimekirjast isomorfsuse kontrolli, tuleb kasutajal joonistada kaks graafi.

GLP pool ülesandest on tunduvalt keerulisem. Ülesande avamisel ilmub paneel, millel on kaks graafi. Graafipaneeli nuppudest saab kasutada vaid tipu liigutamist. Esimese graafi tippudel on ära märgitud tähed. Isomorfismi näitamiseks tuleb teise graafi tippudele lisada samad tähed, mis esimesel graafil, nii et tippude tähistused oleks vastavuses isomorfismiga. (Joonis 8) Teiste sõnadega: Kui f on nende kahe graafi vaheline isomorfism ja v on esimese graafi tipp, siis tipul v ja tipul $f(v)$ peab olema sama täht. Kui kasutaja leiab, et isomorfism on graafidel õigesti märgitud, saab ta vajutada nupule “Kontrolli” ning talle antakse tagasisidet, kas lahendus on õige. Eitaval juhul teavitatakse kasutajat ka tehtud vea kohta. Automaatne kontroll teeb sellisel juhul kindlaks, kas mõlema graafi sama tähega märgitud tippudel on võrdne arv naabreid ja kas nende tippude naabertippudel on samuti samad tähed.

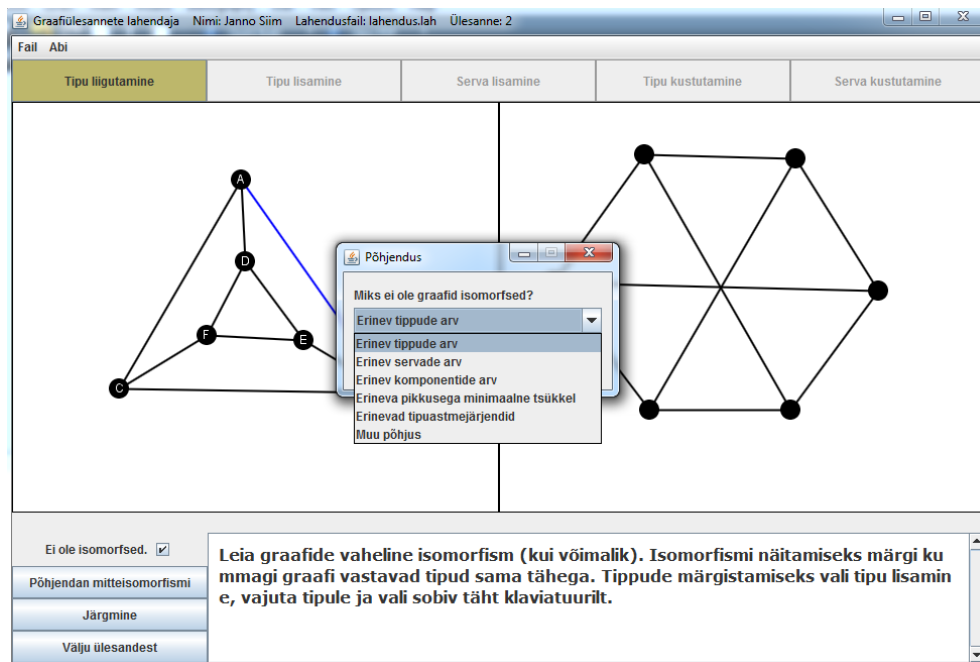
Juhul, kui kasutaja on arvamusel, et graafid ei ole isomorfsed, on tal võimalus teha linnuke märkeruutu nimega “Ei ole isomorfsed”. Selle peale asendub “Kontrolli” nupp uue nupuga “Põhjendan mitteisomorfismi”. Kui kasutaja vajutab sellele nupule, avaneb aken järgnevate graafi invariantidega (Joonis 9):

1. erinev tippude arv;
2. erinev servade arv;
3. erinev komponentide arv;
4. erineva pikkusega minimaalne tsükkel;
5. erinevad tipuastmejärjendid.

Lisaks on veel juures põhjendus “Muu põhjus”, kuna eelnevad invariantid ei moodusta täielikku komplekti, mis suudaks kõiki mitteisomorfseid graafe eristada. Selliste täielike komplektide leidmine on aktiivne uurimisvaldkond graafiteoorias [13] ja elementaarsetest invariantidest seesugust komplekti ilmselt teha ei saakski. Viimast põhjust lubatakse valida ainult sel juhul, kui ükski eelnevatest invariantidest ei ole sobiv.



Joonis 8: Isomorfismi ülesande GLP vaade. Kahe joonisel näidatud graafi vahel on ära märgitud isomorfism.



Joonis 9: Isomorfismi ülesande GLP vaade, kus kasutaja on avanud invariantide akna

Sisult veidi keerulisemad invariantide tingimused, mida kontrollida tuleb, on komponentide arvu ja minimaalse tsükli leidmine. Komponentide arvu leidmiseks kasutatakse graafi süvitsi läbimist, et teada saada, millised tipud on omavahel ühenduses. Sama algortimi kasutati ka eelnevas ülesandes. Minimaalse tsükli pikuse teada saamiseks õnnestus leida järgnev algoritm [14]. Kui põhjenduseks on "Muu põhjus", siis kasutatakse üldist graafide isomorfsuse kontrolli algoritmi VF2, millest oli juttu eelmises peatükis.

Süntaktilised vead:

1. teisel graafil on mõni tipp nimetamata;
2. esimesel või teisel graafil esineb mõni tipu nimi mitu korda;
3. esimesel graafil esineb mingi tipu nimi, mida ei esine teisel graafil või vastupidi.

Sisulised vead:

1. esimesel ja teisel graafil on sama nimega tipp, aga nende tippude naabertippudel ei ole samad nimed;
2. mitteisomorfsuse põhjendus on vale.

Seni realiseeritud ülesandetüüpide seast on üpris kindlasti just sellel tüübil kõige keerukam juhusliku genereerimise viis. Selleks, et isomorfismi tüübi jaoks oleks juhuslik genereerimine mõttekas, peavad genereeritavad graafid vastama kahele tingimusele. Esiteks, kui genereeritud graafid on isomorfsed, siis peaksid nende tipud olema ümber paigutatud nii, et ei oleks koheselt võimalik aru saada, et kaks graafi tõepoolest on isomorfsed. Teiseks, kui genereeritud graafid ei ole isomorfsed, siis peaksid nad olema ikkagi piisavalt sarnased, et lahendajas kahtlust tekitada.

Juhuslik genereerimine võib toimuda viie erineva juhu abil. Tabelis 2 on antud need juhud koos esinemise tõenäosusega. Genereerimise protsessis valitakse esmalt genereerimise juhtum ning seejärel genereeritakse esimene graaf parameetrite järgi, mis on samuti näidatud tabelis 2.

Olgu esimene genereeritud graaf n tipu, m serva ja k komponendiga. Kui varem sai valitud isomorfsuse juhtum, siis genereeritakse teiseks graafiks juhuslikke n tipu ja m servaga graafe seni, kuni esimene ja teine graaf on isomorfsed. Kui sai valitud mõni mitteisomorfsuse juhtumitest, siis genereeritakse teiseks graafiks juhuslik graaf, millel on täidetud tabelis 2 näidatud tingimused. Tabeli täpsustuseks

tuleks mainida, et pluss-miinus lahtrites valitakse märk võrdse tõenäosusega juhuslikult. Tühjad lahtrid tähendavad, et vastav väärtus võib olla suvaline. Võib tekkida olukorda, kus teise graafi jaoks on parameetrite täitmine võimatu. Sellises olukorras alustatakse kogu juhusliku genereerimise protsessi algusest peale.

Tabel 1: Esimese graafi juhusliku genereerimise parameetrid koos juhtumite tõenäosustega

	Esinemise tõenäosus	Tippude arvu alampiir	Tippude arvu ülempiir	Servade arvu alampiir	Servade arvu ülempiir
Isomorfsed	$\frac{1}{2}$	5	8	30%	80%
Mitteisomorfsed juhtum 1	$\frac{1}{12}$	6	9	30%	80%
Mitteisomorfsed juhtum 2	$\frac{1}{12}$	5	7	30%	75%
Mitteisomorfsed juhtum 3	$\frac{1}{12}$	8	12	10%	30%
Mitteisomorfsed juhtum 4	$\frac{1}{4}$	5	7	30%	80%

Tabel 2: Teise graafi juhusliku genereerimise parameetrid. Parameetrid n , m ja k tähistavad vastavalt esimese graafi tippude, servade ja sidusate komponentide arvu.

	Tippude arv	Servade arv	Komponentide arv
Mitteisomorfsed juhtum 1	$n \pm 1$	m	-
Mitteisomorfsed juhtum 2	n	$m \pm 1$	-
Mitteisomorfsed juhtum 3	n	m	ei võrdu k -ga
Mitteisomorfsed juhtum 4	n	m	k

4.4 Naabusmaatriksile vastava suunatud graafi joonistamine

Selles ülesandetüübis tuleb naabusmaatriksi järgi konstrueerida suunatud graaf. Sarnane tüüp on olemas ka suunamata graafide jaoks.

Kõigepealt tuleb juttu ülesande koostamise programmist. GKP-s avaneb selle ülesande valimise puhul nii graafipaneel kui ka maatriksipaneel. Esiteks tuleks kasutajal ära täita temale sobivat järku maatriks. Seejärel on võimalus anda ette ka mingi osa suunatud graafist. Kontrollitakse, kas teostatud graafi osa on vastavuses maatriksiga. Vastasel juhul kuvatakse veateade.

Lahendamise programmis tuleb õpilasel joonistada ette antud naabrusmaatriksi järgi suunatud graaf või täiendada olemasolevat graafi nii, et see vastaks naabrusmaatriksile. Ülesande kontrollimine on seejuures üpriski ootuspärane. Programm teeb kindlaks, kas graafil on õige arv tippe ja neil maatriksile vastavad tähed ning kas graafil on täpselt need kaared, mis seal naabrusmaatriksi järgi peaksid olema.

Süntaktilised vead:

1. mõni tipp graafil on nimetamata;
2. korduv tipu nimi graafil.

Sisulised vead:

1. graafil on vale tippude arv;
2. graafi tippude nimed ei ühti maatriksil märgitud tippude nimedega;
3. graafil on liigne kaar;
4. graafilt on mõni kaar puudu;
5. graafi kaared on tippudega valesti ühendatud.

Ülesande juhuslikuks genereerimiseks valitakse ühtlase jaotusega vahemikust 3 kuni 5 maatriksi järk n . Ühtede arv m maatriksil valitakse ühtlase jaotusega vahemikust 40% kuni 70% n^2 -st. Arv n^2 tuleb siin sellest, et n -tipulisel suunatud graafil saab olla makimaalselt n^2 serva. Valitud m ühte paigutatakse juhuslikele positsioonidele maatriksis ning ülejäänud osa maatriksist täidetakse nullidega.

4.5 Suunatud graafide vastava naabrusmaatriksi konstrueerimine

See ülesandetüüp kujutab endast eelmise tüübi pöördülesannet. Ette antud suunatud graafi põhjal tuleb õpilasel konstrueerida naabrusmaatriks. Ka sellel ülesandetüübil leidub suunamata graafide analoog.

Koostamise programmis avanevad selle ülesande valimisel graafipaneel ja maatriksipaneel täpselt nagu eelmises ülesandetüübis, kuid selle ülesande puhul tuleb koostajal joonistada graaf ning maatriks võib olla osaliselt täidetud. Taaskord kontrollitakse, kas osaline naabusmaatriksi kirjeldus vastab joonistatud graafile.

Lahendamise programmis tuleb õpilasel konstrueerida maatriks või täiendada varem olemas olevat maatriksit nii, et see vastaks etteantud suunatud graafile. Kontrollimisel tehakse kindlaks, kas graafi tippude arv ja maatriksi järk on võrdsed ning kas neil on samad tähed. Seejärel kontrollitakse, kas naabusmaatriksil on ära märgitud täpselt need kaared, mis antud graafis esinevad.

Süntaktilised vead:

1. mõni maatriksi rida/veerg on nimetamata;
2. korduv rea/veeru nimi maatriksil;
3. mõni maatriksi lahter on tühi.

Sisulised vead:

1. vale maatriksi järk;
2. maatriksi rea/veeru nimed ei ühti graafi tippude nimedega;
3. maatriksis on liigne "1";
4. maatriksis on liigne "0";
5. maatriksis on üks "1" või "0" vales asukohas.

Ülesande juhusliku genereerimise jaoks valitakse võrdse tõenäosusega tippude arvuks n arv 4 või 5 ning servade arvuks m valitakse 30% kuni 50% n^2 -st. Parameetrid on tavapärasest väiksemad, kuna suunatud graafil on maksimaalne servade arv märgatavalt suurem. Seejärel genereeritakse juhuslik n tipu ja m servaga graaf.

Kokkuvõte

H. Pärn tegi 2013. oma bakalaureuse töö raames prototüübi graafiteooria õpiprogrammist, millega oleks võimalik koostada ja lahendada erinevaid graafiteooriaga seotud ülesandeid. Antud lõputöö eesmärgiks sai seatud selle prototüübi edasiarendamine ja keerukamate ülesandetüüpide sisse toomine.

Ülesandetüübid, mis said programmi lisatud:

- regulaarsete graafide joonistamine
- sidusate komponentidega graafi joonistamine
- isomorfsuse kontrollimine
- naabrusmaatriksile vastava suunatud graafi joonistamine
- suunatud graafile vastava naabrusmaatriksi konstrueerimine

Samuti sai tehtud mitu üldist täiendust, mis on ilmselt väga olulised selle tarkvara edasiarendamiseks. Neist olulisimana võiks välja tuua:

- kõigi ülesandetüüpide juhusliku genereerimise võimalus ja üldised meetodid programmis kasutatavate andmestruktuuride juhuslikuks genereerimiseks
- võimalus suure hulga graafide samaaegseks haldamiseks (mitme graafi paneel)
- graafide isomorfsuse kontrollimise võimaluse sisse toomine
- suunatud graafide esituse võimalus

Sai ka antud ülevaade suurest hulgast graafiteooria ülesandetüüpidest, mida saaks selles tarkvaras kasutada. Siinjuures vaadati, et ülesandetüübid vastaks ka

aines DME õpetatavale graafiteooria materjalile. See ülesandetüüpide nimekiri annab hea aluse antud tarkvara edasiarenduseks. Kuna selgus, et sobilikke ülesandetüüpe on päris suur hulk, siis autori hinnangul oleks edasiarenduse jaoks tööd veel minimaalselt ühe või kahe bakalaureusetöö jagu. Siis peaks saama kaetud vähemalt kõige tähtsamad ülesandetüübid.

Computerization of Graph Theory Exercises

Bachelor's thesis (6 ECP)

Janno Siim

Summary

In 2013 in the context of his bachelor's thesis H. Pärn developed a prototype for an educational software that would allow to create and solve different types of exercises related to graph theory. Current thesis sets its goal to develop that prototype and to introduce new and more complex exercise types.

Exercise types that were added to the software:

- drawing regular graphs
- drawing graphs with connected components
- isomorphism checking
- transferring graph from adjacency matrix representation to a drawing of a directed graph
- transferring a drawing of a directed graph to adjacency matrix representation

Also, couple of general additions were made that will probably turn out to be important in developing this software. Perhaps some of the most significant among them are the following:

- possibility of random exercise generation for all current exercise types and general methods that would allow randomly generating data structures that are used in this software
- panel that would allow working with several graphs at the same time
- introduction of a graph isomorphism algorithm
- directed graph representation

In addition, an overview of a huge number of graph theory exercises were given that could be applied to this software. Such a list of exercise types gives a perspective how to move on with the development. As it was discovered that there are a lot of exercise types that could be added, author feels that there is probably enough work for at least the amount of one or two bachelor's thesis before the most important of the types will be covered.

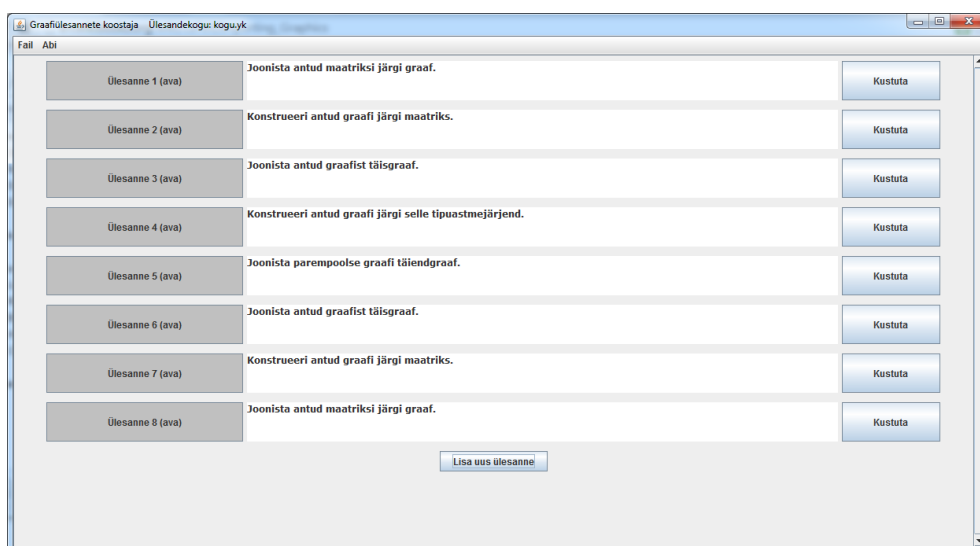
Kirjandus

- [1] H. Pärn. *Graafiülesannete lahendamise keskkond*, 2013.
- [2] M. Aigner, E. Triesch. *Realizability and uniqueness in graphs*. — Discrete Mathematics, 1994, 136 (1-3), 3-20.
- [3] R. Palm. *Diskreetse matemaatika elemendid*, Tartu Ülikooli Kirjastus, 2003.
- [4] A. Buldas, P. Laud, J. Willemson. *Graafid*, Tartu Ülikooli Kirjastus, 2008.
- [5] J. Matoušek, J. Nešetřil. *Invitation to Discrete Mathematics*, Oxford University Press, 2011.
- [6] O. Ore. *Graafid ja nende kasutamine*, Valgus, 1976.
- [7] C. Boucher, D. Loker. *Graph Isomorphism Completeness for Perfect Graphs and Subclasses of Perfect Graphs*. — 38th Southeastern International Conference on Combinatorics, 2006.
- [8] A. Jovanovič, D. Danilocič. *A New Algorithm for Solving the Tree Isomorphism Problem*. — Computing, 1982, 32 (3), 187-198.
- [9] J. E. Hopcroft, J. K. Wong. *Linear time algorithm for isomorphism of planar graphs*. — Proceeding STOC '74 Proceedings of the sixth annual ACM symposium on Theory of computing, 1974, 172-184.
- [10] E. M. Luks. *Isomorphism of Graphs of Bounded Valence Can Be Tested in Polynomial Time**. — Journal of Computer and System Sciences, 1982, 25 (1), 42-65.
- [11] L. P. Cordella, P. Foggia, C. Sansone, M. Vento. *An Improved Algorithm for Matching Large Graphs*. — 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, 2001, 149-159.
- [12] N. J. A. Sloane. *Number of regular graphs with n nodes*, 2012
<http://oeis.org/A005176> – viimati vaadatud 20.04.2014.

- [13] M. Dehmer, F. Emmert-Steib, M. Grabner. *A computational approach to construct a multivariate complete graph invariant.* — Information Sciences, 2013, 260, 200-208.
- [14] webcourse.cs.technion.ac. *Computing the girth of a graph*, 2003
<http://webcourse.cs.technion.ac.il/234247/Winter2003-2004/ho/WCFiles/Girth.pdf> – viimati vaadatud 30.04.2014.

Lisa 1

Ekraanitõmmised



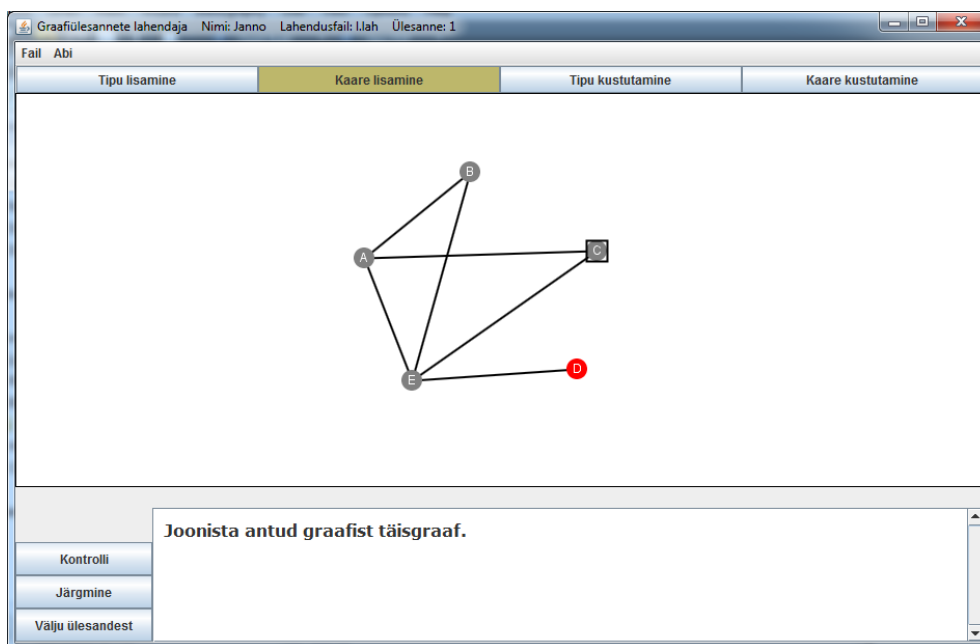
Joonis 10: GKP ülesannete nimekirja paneel



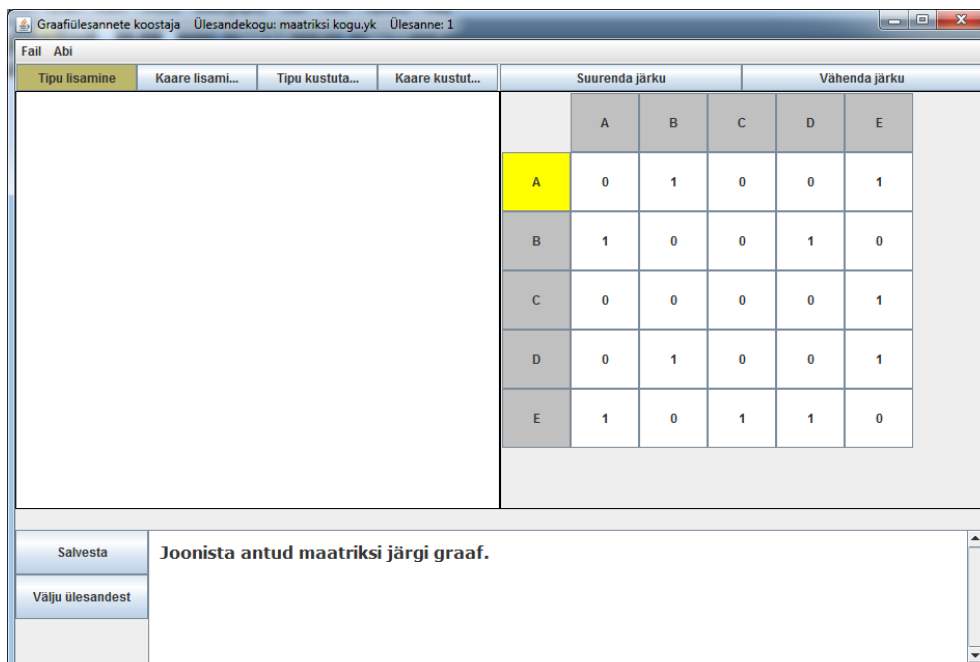
Joonis 11: GLP ülesannete nimekirja paneel

Graafiülesannete koostaja		
Fail Abi		
Nimi: Janno		Lahendusfail: lahendus.lah
		Kontrollitavate lahenduste arv
Ülesanne 1 - õigesti	Lahendus 1 (+)	Süntaksiviga: Mõni graafi tipp on nimetamata.
Ülesanne 2 - valesti	Lahendus 1	Viga: Maatriksis on liigne "kaar"!
		Viga: Maatriksis on liigne "kaar" (2)
		Viga: Maatriksi rea/veeru nimed ei ühti graafi tipunimedega!
		Viga: Maatriksis on liigne "kaar" (3)
Ülesanne 3 - õigesti	Lahendus 1 (+)	-
Ülesanne 4 - õigesti	Lahendus 1 (+)	Süntaksiviga: Mõni tipuastmejärjendi lahter on tühi!
		Süntaksiviga: Tipuastmejärjend ei ole mittekasvatav!
		Viga: Viga tipujärjendis!
Ülesanne 5 - õigesti	Lahendus 1 (+)	Viga: Vale tippude arv graafil!
		Süntaksiviga: Mõni tipp graafil on nimetamata.
Ülesanne 6 - õigesti	Lahendus 1 (+)	-
Ülesanne 7 - õigesti	Lahendus 1 (+)	-
Ülesanne 8 - õigesti	Lahendus 1 (+)	-

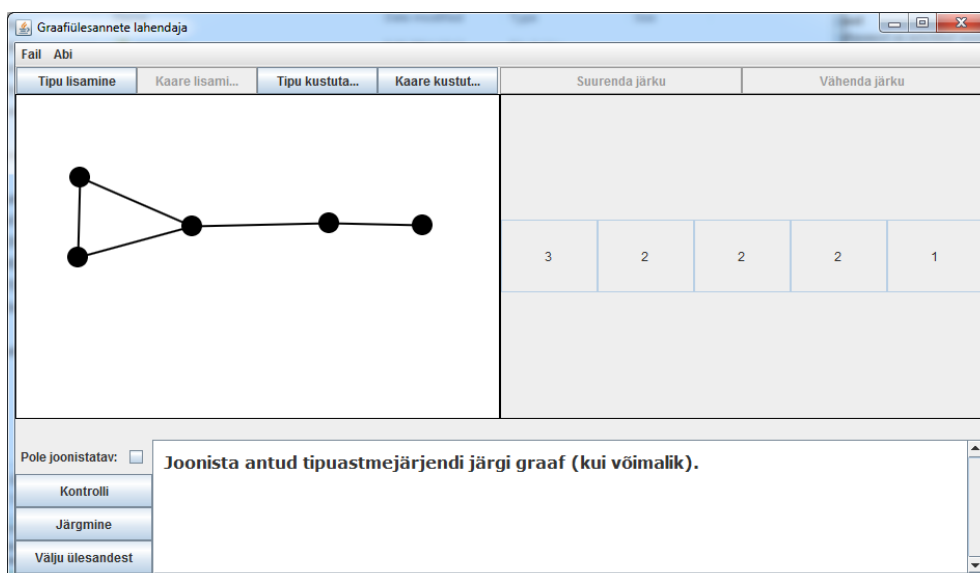
Joonis 12: GKP lahenduste kontrolli paneel



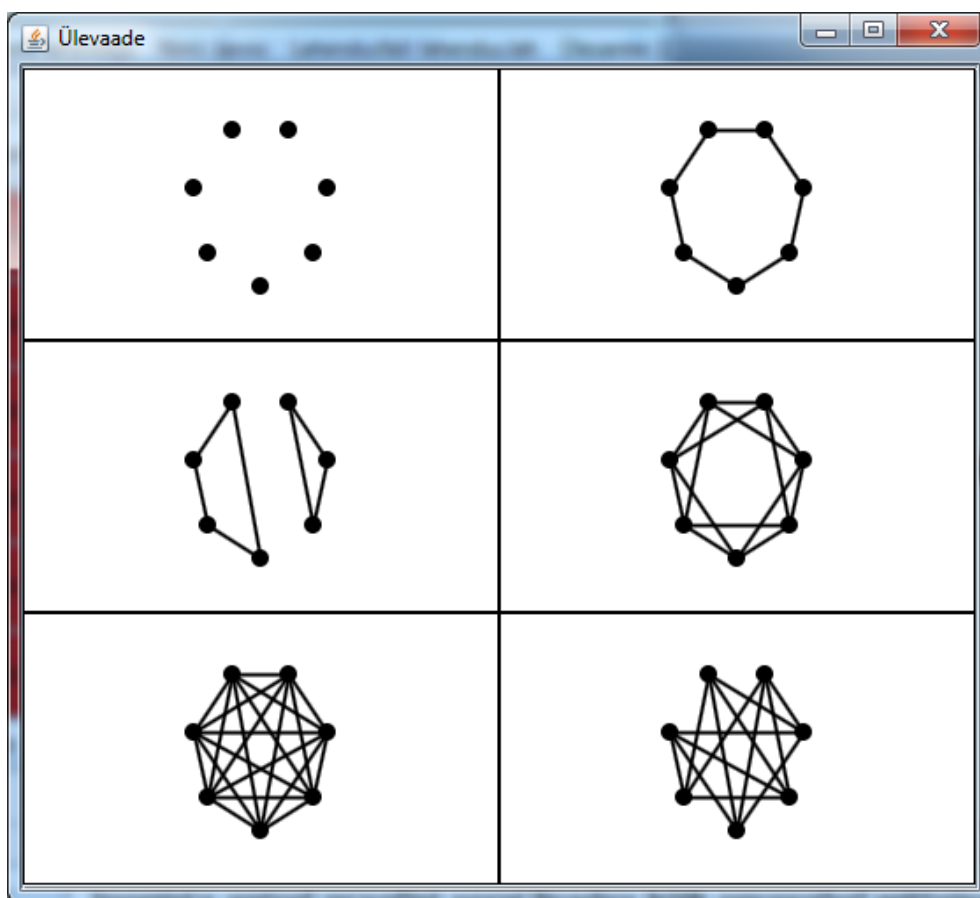
Joonis 13: Joonisel on näha graafipaneeli täisgraafi moodustamise ülesande lahendamisel



Joonis 14: Joonisel paremal poolel on näha maatriksi paneeli



Joonis 15: Joonisel paremal poolel on näha tipuastme paneeli



Joonis 16: Mitme graafi paneeliga kaasa tulev ülevaate aken. Hetkel on nähtaval kõik 7-tipulised regulaarsed graafid.

Lisa 2

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Janno Siim (sünnikuupäev: 18.03.1992)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Graafiteooria ülesannete kompileerimise, mille juhendaja on Rein Prank,
 - 1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 29.04.2014